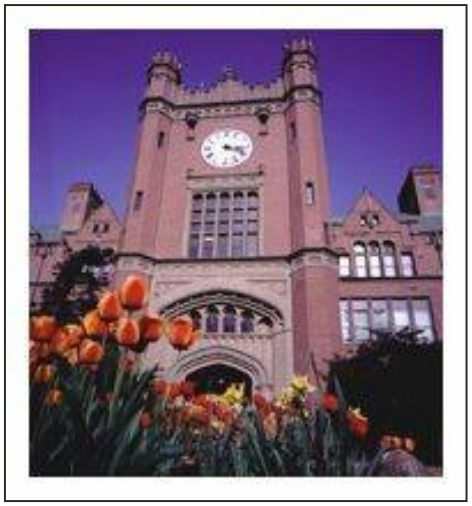
MILS, Multiple Independent Levels of Security: A High Assurance Architecture



Carol Taylor
University of Idaho
Moscow, Idaho

Outline

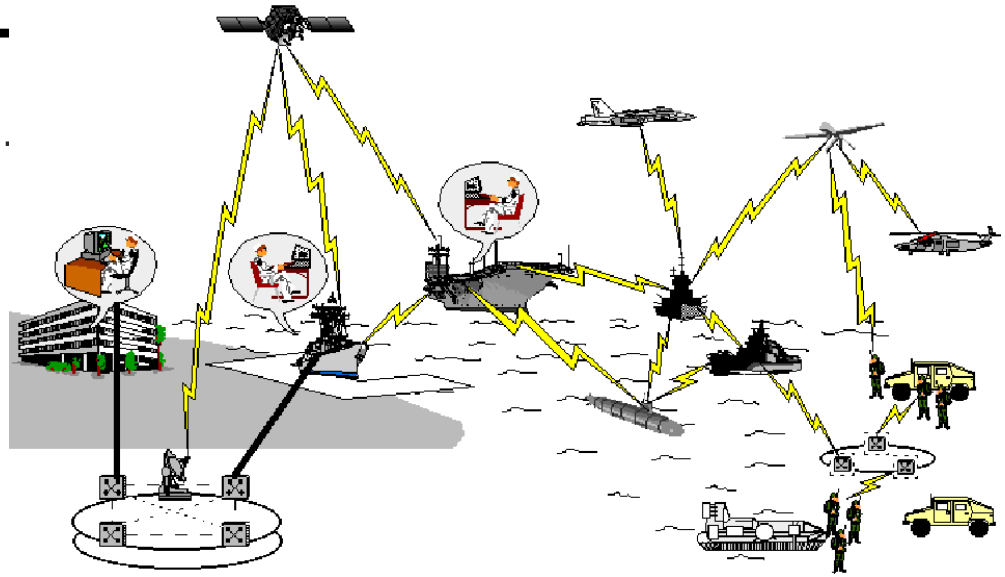
- Need for MILS
- MILS Architecture
  - Separation Kernel
  - Middleware
  - Applications
- MILS Security Policies
- Certification
- Progress
- Future Research

Need for MILS

- DOD has a long standing need for Multi-level Secure (MLS) systems
  - Limit access to information by different classification levels
    - Unclassified, Confidential, Secret, Top Secret
  - Most concerned with confidentiality but integrity also important

Need for MILS

- Modern warfare is about sharing informa
- Information must be shared securely to not compromise the mission



- Information is rapidly becoming more diverse
  - Coalition Force Operations
  - Multiple Levels and Communities of Interest
  - Smart Push / Smart Pull / Web Services
  - True MSLS/MLS capability is becoming more important

Need for MILS

- Systems in 70’s and 80’s
  - Mostly mainframes, standalone
  - Limited physical access
- Security concepts at that time
  - Segregate security functionality from rest of system
    - Reference Monitor
    - Security Kernel

Need for MILS

- Reference Monitor – What is it?

Need for MILS

- Reference Monitor – What is it?
  - Access decisions go through a small software or software and hardware monitor
  - Uses an access control database or list
  - Is non-bypassable, tamperproof and small so correctness is easy to verify
    - Formally

Need for MILS

- Security Kernel – What is it?

Need for MILS

- Security Kernel – What is it?
  - Set of security enforcement mechanisms that implements Reference Monitor concept
  - Segregated security functionality that implements system security policy
  - Implements access control and possibly multi-level access decisions

Need for MILS

- Security Kernels Not Ideal
- As security policy became more complex:
  - Code grew in security kernel
  - Certification efforts become unmanageable
  - Evaluatability of kernel decreased
  - Maintainability of kernel code decreased
  - Single Security Policy made security difficult to use and hard to enforce

MILS Architecture

MILS Architecture

- MILS is an evolving layered, component based high assurance architecture
  - MILS = Multiple Independent Levels of Security
- Under development by industry, government and academia
  - Vendors developing separation kernels
    - Green Hills, LynuxWorks, Wind River
  - Others developing MILS components
    - Lockheed Martin, Objective Interface, University of Idaho, Naval Research Lab
- Intended for high assurance environments
  - Multi-level data communications
  - Safety critical systems

MILS Architecture

- Dramatically reduce the amount of *security critical code*

So that we can

- Dramatically increase the scrutiny of *security critical code*

To make

- Development, certification, and accreditation more practical, achievable, and affordable.

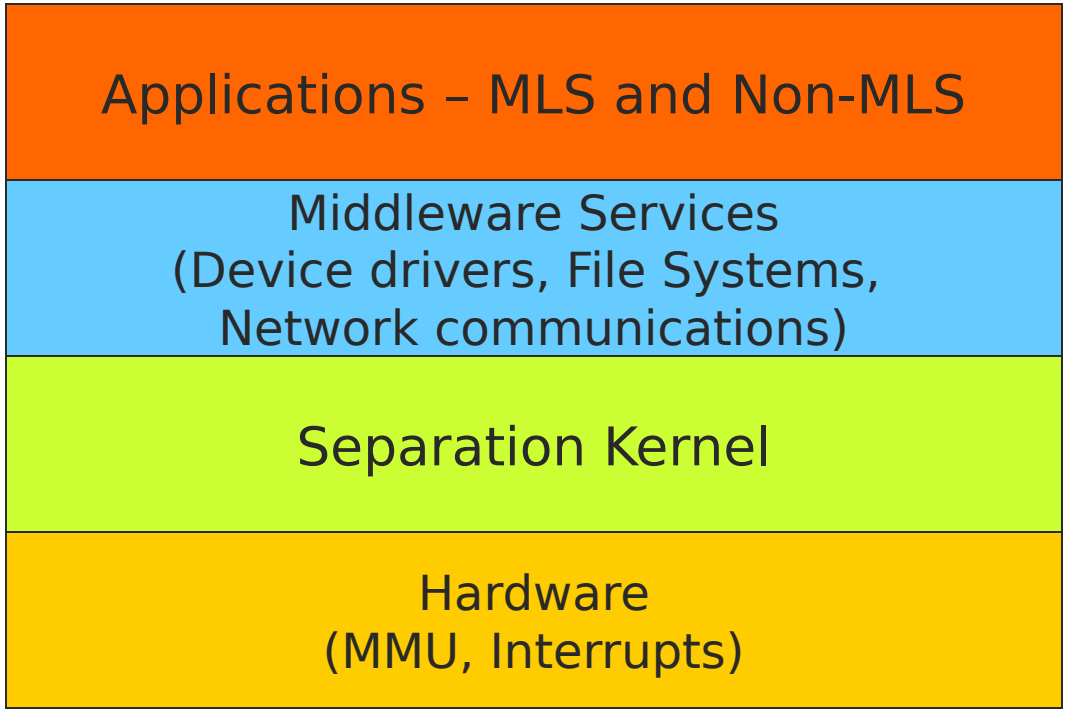
MILS Architecture

- Layered Architecture
  - Separation Kernel, Middleware, Application
- Different concept than traditional security kernel based architectures
- Security policy
  - Multiple policies differ between layers
  - Assume applications have their own policies
  - Offer services for application security

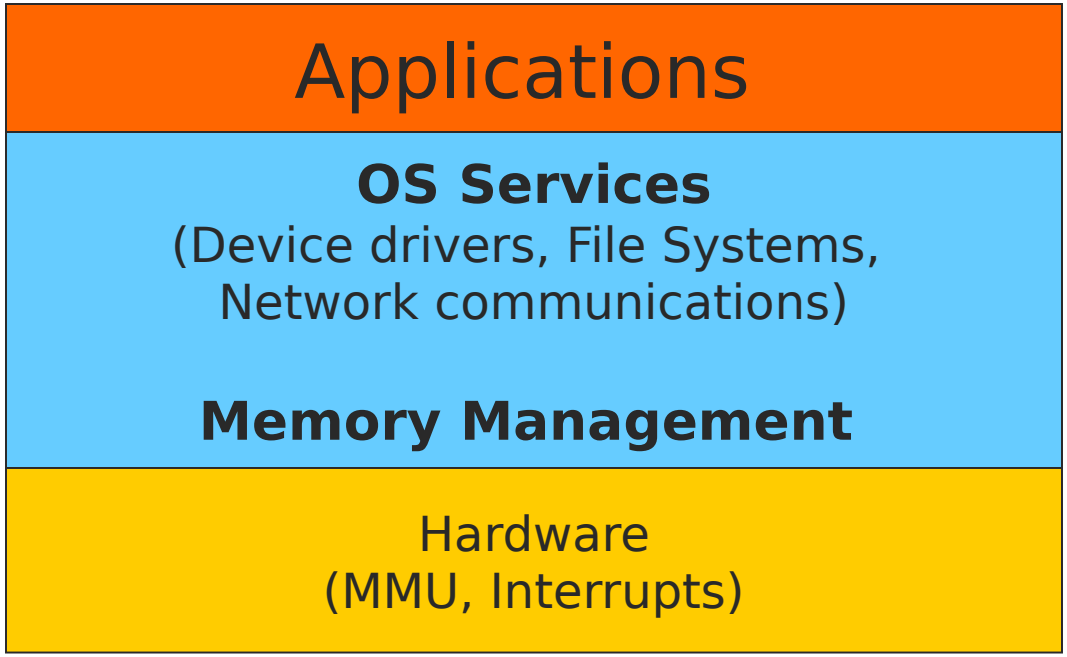
MILS Architecture

- Separation Kernel
  - Separate process spaces (partitions)
  - Secure transfer of control between partitions
  - Really small: 4K lines of code
- Middleware
  - Extends the usual concept of middleware
  - Contains OS components traditionally found in kernel
    - Device Drivers, File Systems, Network Stacks
- Applications
  - Implement application-specific security functions
    - Firewalls, Cryptomodules, Guards, User applications

MILS Architecture View



Normal Architecture View



Separation Kernel

- Time and Space Partitioning
  - Data Isolation
  - Information Flow
- Multi-Threading
- Inter-Partition Communication
  - OS dependent, shared memory, Arinc 653 channels
- Resource Sanitization system registers
- Minimum Interrupt Servicing
- Partition Scheduler

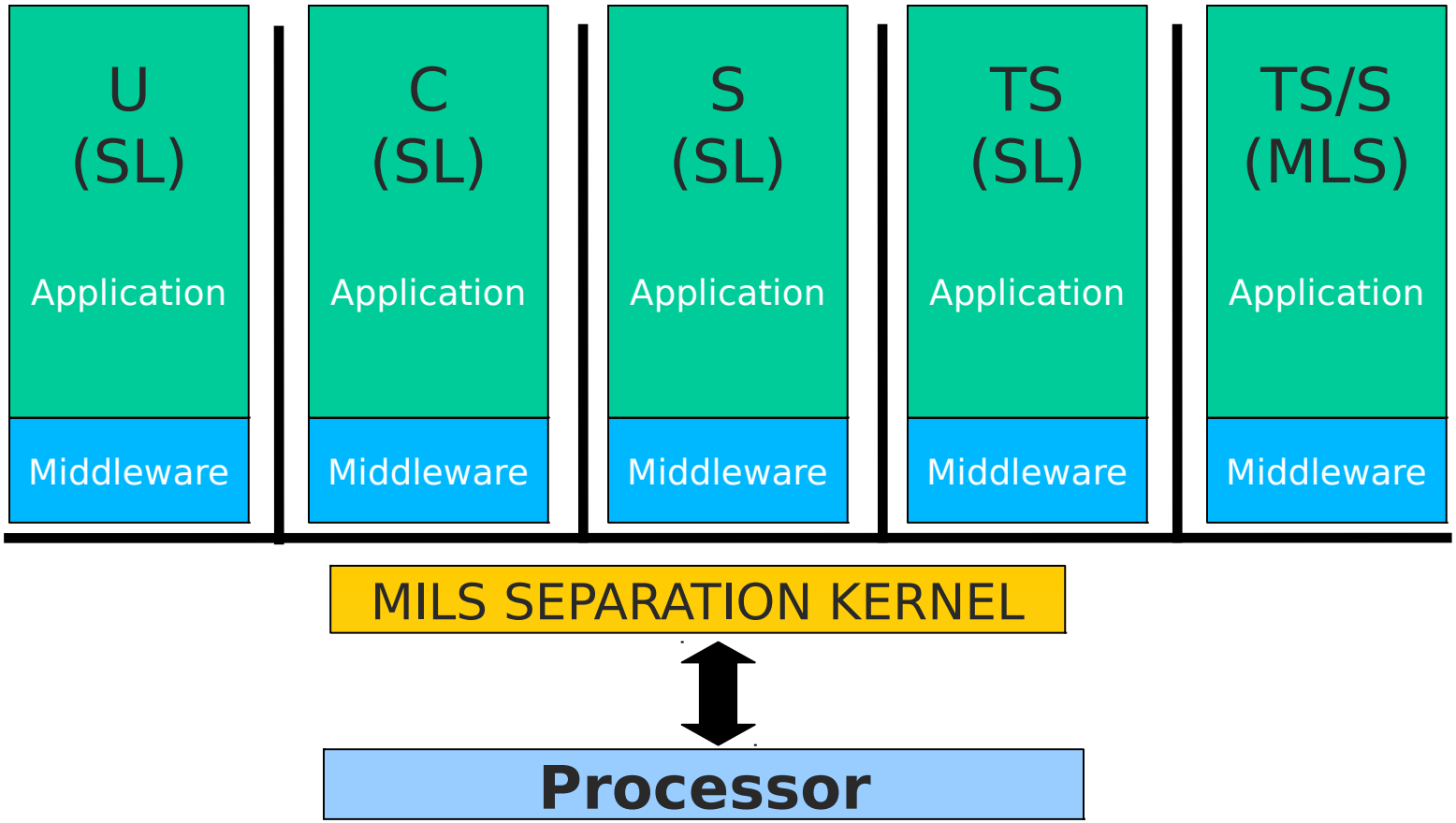
Middleware

- Traditional RTOS Services
  - Device Drivers
  - File Systems and access to shared devices
  - Network communication
- Guarded Communication System
  - Partitioning Communication System
    - Inter-Machine Communication
      - Access Control and Encryption across the network
    - Quality of Service
      - TCP, UDP, Firewire, …

Applications

- Traditional Middleware
  - CORBA (Distributed Logic)
  - DDS (Distributed Data, Smart Push)
- Other Applications
  - Cross Domain Services
    - Access Guards
    - Content Guards (.xml, .doc, .pdf)
    - Inter-domain Guards (legacy systems)

MILS Architecture



MILS Architecture

- MILS makes mathematical verification of core systems and communications software possible
  - Reduces security functionality to four key security policies
    - Information Flow
    - Data Isolation
    - Resource Sanitization
    - Damage Limitation

MILS Security Policies

MILS Security Policies

- **Separation Kernel Policy**
- Information Flow
  - Information originates only from authorized sources
  - Information is delivered only to intended recipients
  - Source of Information is authenticated
- Data Isolation
  - Information in a partition is accessible only by that partition
  - Private data remains private
- Resource Sanitization
  - The microprocessor itself will not leak information from one partition to another as it switches from partition to partition
- Damage Limitation
  - A failure in one partition will not cascade to another partition
  - Failures will be detected, contained, & recovered from locally

MILS Security Policies

- **Middleware Policies**
- Partitioning Communication System (PCS)
  - PCS is communications middleware for MILS
  - Interposes inter-node communications
  - Partitions Network communication between processors
  - Allows partitions to communicate over a network
  - Deals with encryption and Bandwidth provisioning

MILS Security Policies

- **Application Policy**
- Access Guards
  - Protocol Specific Access Control
    - CORBA/GIOP (Client/Server) Access Guard
      - Determines if query is allowed based on method name, parameter values, security levels of client/server
      - Determines if response is expected
      - Error Message Response Policy
    - DDS (Publish/Subscribe) Access Guard
      - Determines if subscriber allowed to connect/receive from a particular label based on identity and security levels of label and subscriber
      - Determines if publisher allowed to connect/publish to a particular label based on identity and security levels of label and publisher
    - HTTP (Web) Access Guard

MILS Security Policies

- Separation Kernel
  - Focuses on partitions
  - Authorized communication between partitions
- Middleware – PCS
  - Authorized communication across a network
- Application – GIOP CORBA Guard
  - Authorized delivery of messages between processes using the CORBA protocol

Certification

Certification

- Common Criteria 2.2
  - Certification of single products
    - Application, OS, processor
  - Target of Evaluation (TOE)
  - Certification Process
    - 1. Define or find a Protection Profile (PP)
    - 2. Adapt PP to a Security Target (ST) at a EAL level
      - ST specifies security functionality of TOE
    - 3. Evaluated according to ST
      - NIAP Lab evaluates products up to EAL 4
      - Beyond EAL 4, NSA evaluates TOE

Certification

- Common Criteria 3.0
  - Allows certification of composed products
    - Involves combination of two or more evaluated products
    - Intent is to evaluate components developed by different organizations
      - Proprietary issues
      - Assumption is not all information is available for evaluation

Certification

- Composed CC 3.0 Certification
  - How to do it?
    - 1. Independent evaluation of each component
    - 2. Composed evaluation, identify base component and dependent component
    - 3. Use ACO: Composition - Five families
      - ACO-COR – Composition rationale
      - ACO-DEV – Development evidence
      - ACO-REL – Reliance of dependent component
      - ACO-TBT – Base TOE Testing
      - ACO-VUL – Composition vulnerability analysis

Certification

- Composed CC 3.0 Certification cont.
  - 3. ACO: Composition
    - Insures base component provides at least as high an assurance level as the dependent component
    - Insures that security functionality of base component in support of dependent component is adequate
    - Provides for a description of interfaces used to support security functions of dependent component
      - May not have been considered during component evaluation

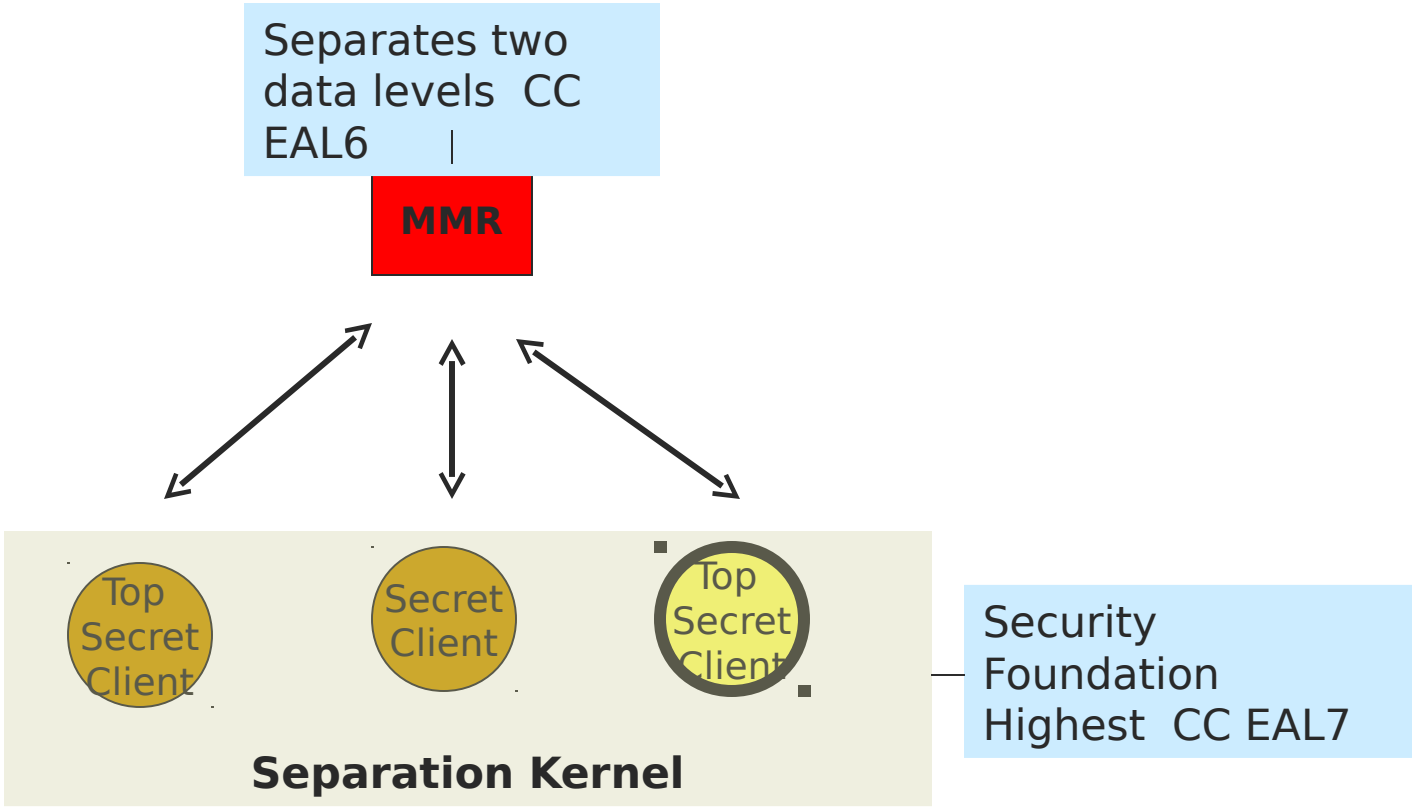
Certification

- MILS Certification
  - MILS is ideally suited to a composed certification effort
    - MILS was designed as a component architecture
      - Components designed by multiple vendors
      - Components certified at multiple EAL levels
      - Components assist with security policy enforcement

Certification of MILS Components

- Composed MILS CC Certification
  - **Example**: Separation Kernel and MMR
    - Base component
      - Separation Kernel
    - Dependent component
      - MILS Message Router (MMR)
        - Software router for MLS data
        - Has its own security policy

Certification of MILS Components



Certification of MILS Components

- Steps for Composing MILS Components
  1. Evaluation of Separation Kernel
  2. Develop ST for MMR no PP available
  3. Evaluation of Composed MILS Components
     - Define ST for composed system
     - Develop artifacts for composed system
       - Formal Security policy
       - Other documents
     - Evaluate Composed system

MILS Progress

MILS Progress

- **Separation Kernel and components**
  - Green Hills kernel complete and under evaluation
  - Lynux Works and Wind River – kernels under development
  - PCS under development by Objective Interface
  - University of Idaho developing GIOP Guard, MMR
  - All vendors are developing solutions for MILS Workstations – currently “demoware”
- **Certification Efforts**
  - Separation Kernel PP, Security Target - done
    - Currently being evaluated
      - Target EAL 6+
  - Composition example - University of Idaho
    - MMR or GIOP Guard, MMR and Separation Kernel

Future Research

Future Research

- Need a ***Calculus of Assurance Composition***
  - Not yet provided by research/scientific community
- Need to define objectives for a Formal Integration Framework
  - Term coined by Rance DeLong

Future Research

- **Formal Integration Framework**
- Preserve component properties in composition
- Allow independent development of interoperable components
- Shift emphasis from component to framework
- Have interface assumptions and service guarantees
- Put “Firewalls” between components
- Provide composability guarantees for “well-behaved” components
- Model communications among components

Future Research

- Performance Studies
  - Safety world, very limited partitions and no extra security
  - MILS world many more context switches
  - Concern is that MILS may be too slow for some applications
  - Need more performance data using all MILS components to evaluate speed and timing issues
  - Many prospective applications have real-time constraints
- University of Idaho engaged in this work

Questions?

End